

Background

Partnet supports its agile application development processes by maintaining a flexible, continuous delivery environment enabled by a number of open source technologies. A number of these assets existed prior to this solicitation, namely:

- Packer configuration and pre-generated base images
- Jenkins continuous integration server
- Artifactory - versioned binary repository
- Ansible component library

For the pool-2 prototype, consistent development environments were rapidly provisioned for our team using a Packer generated CentOS base image running on the VirtualBox platform.

To ensure consistency between the virtual development environments and our "production" deployment to Amazon's EC2 system, the Ansible (<http://www.ansible.com/home>) system is used to provision the environments and orchestrate deployments. This combination of tools ensures new developers can be introduced to the project with a fully-functional development environment in minutes that is consistent in form and function with our other secondary environments (test, stage, production). To limit the repository to only what was relevant to the prototype, elements of our common ansible configuration were copied. Additional work was necessary to test the reconfiguration, project specific properties, etc.

A full production continuous delivery pipeline requires a number of tools beyond a VCS system, particularly in an environment that wishes to keep from bloating the source code repository with versioned binary artifacts. Specifically, Partnet uses a continuous integration server, Jenkins, and a binary artifact repository, Artifactory. Jenkins is responsible for monitoring the source code repository for changes, generating automated builds, running the associated automated test suites, and publishing to Artifactory. Artifactory maintains versioned binary build artifacts as well as 3rd party libraries distributions and installation artifacts (e.g. Java, Elastic Search, WildFly, etc). Ansible stages the necessary artifacts from Artifactory to the control host before deploying these to the target environment.

For the prototypes, a new project was configured in Jenkins with bindings to the Github repository and Partnet's Artifactory instance. On change, Jenkins will sync to the repository, reprovision a new virtual environment, performing an automated Gradle build and ansible deployment, run automated end-user browser testing against it, and, if successful, publish the build artifacts to Artifactory.